Faculty
Leo Bachmair, Professor, Ph.D., University of Illinois, Urbana-Champaign: Computational logic; automated deduction; symbolic computation.
Hussein G. Badr, Associate Professor, Ph.D., Penn State University: Computer communication networks and protocols; performance evaluation, modeling and analysis.
Michael A. Bender, Associate Professor, Ph.D., Harvard University: Algorithms; scheduling; asynchronous parallel computing.
Arthur J. Bernstein, Professor Emeritus, Ph.D., Columbia University: Transaction processing; concurrent programming; distributed databases.
Tzi-ckrer Chieu, Professor, Ph.D., University of California, Berkeley: Processor architecture; parallel I/O; high-speed networks; compression.
Samir Das, Associate Professor, Ph.D., Georgia Institute of Technology: Wireless networking and mobile computing; parallel and distributed simulation; performance evaluation of computer systems and networks.
Ahmad Esmaili, Lecturer, M.S., Stony Brook University: Business and scientific applications of computing; relational database management systems.
Jie Gao, Assistant Professor, Ph.D., Stanford University: Wireless networking.
Herbert L. Gelernter, Professor Emeritus, Ph.D., University of Rochester: Artificial intelligence; knowledge-based, heuristic problem-solving systems; scientific applications.
Radu Grosu, Assistant Professor, Ph.D., Technical University, Munich: Software and systems engineering; design automation for embedded systems; applied formal methods.
Xianfeng Gu, Assistant Professor, Ph.D., Harvard University: Computer graphics.
Himanshu Gupta, Assistant Professor, Ph.D., Stanford University: Information systems, databases.
Arie Kaufman, Distinguished Professor, Ph.D., Ben Gurion University, Israel: Computer graphics; visualization; virtual reality; user interfaces; multimedia; computer architecture.
Robert F. Kelly, Lecturer, Ph.D., New York University: Information systems; software engineering; electronic commerce; parallel programming.
Michael Kifer, Professor, Ph.D., Hebrew University of Jerusalem: Database systems; logic programming; knowledge representation; artificial intelligence.
Ker-I Ko, Professor, Ph.D., Ohio State University: Computational complexity; theory of computation; computational learning theory.
Philip M. Lewis, Professor Emeritus, Ph.D., Columbia University: Concurrency and concurrent systems; transaction processing systems; software engineering.
Y. Annie Liu, Associate Professor, Ph.D., Cornell University: Programming languages and compilers; program optimization; program analysis and transformation; programming environments; reactive systems; algorithm design.
Richard McKenna, Lecturer, M.S., Stony Brook University: Java programming; image processing.
Alexander Mohr, Assistant Professor, Ph.D., University of Washington: Robust and scalable delivery of time-sensitive multimedia and incorporate elements of algorithms, networks, and multimedia systems.
Klaus Mueller, Associate Professor, Ph.D., Ohio State University: Visualization; computer graphics; medical imaging; image-based rendering; virtual reality; distributed virtual environments.
Theo Pavlidis, Professor Emeritus, Ph.D., University of California, Berkeley: Image processing; machine vision; computer graphics; window systems.
Shaunak Pawagi, Lecturer, Ph.D., University of Maryland, College Park: Analysis of algorithms; parallel computing.
Hong Qin, Professor, Ph.D., University of Toronto: Computer graphics; geometric modeling and design; physics-based animation and simulation; scientific computing and visualization; virtual environment; computer vision; medical imaging; applied mathematics.
C.R. Ramakrishnan, Associate Professor, Ph.D., Stony Brook University: Logic programming; programming languages; verification.
I.V. Ramakrishnan, Professor, Ph.D., University of Texas, Austin: Computer architecture; algorithms; rewrite systems.
Dimitris Samaras, Associate Professor, Ph.D., University of Pennsylvania: Computer vision; computer graphics; medical imaging; animation and simulation; image-based rendering; physics-based modeling.
Anthony Scarlatos, Lecturer, M.A., Adelphi University: Multimedia; interface design; computer-based training; distance learning.
R. Sekar, Professor, Ph.D., Stony Brook University: Computer and network security; software/distributed systems; programming languages; software engineering.
Radu Sion, Assistant Professor, Ph.D., Purdue University: Computer security.
Steven Skiena, Professor, Ph.D., University of Illinois, Urbana-Champaign: Algorithms; computational biology; computational geometry.
David R. Smith, Professor Emeritus, Ph.D., University of Wisconsin, Madison: Hardware description languages and synthesis; VLSI design tools; experimental chip architectures.
Scott A. Smolka, Professor, Ph.D., Brown University: Model checking; semantics of concurrency; CASE tools for safety-critical systems; distributed languages and algorithms.
Eugene W. Stark, Professor, Ph.D., Massachusetts Institute of Technology: Programming language semantics; distributed algorithms; formal specifications; verification; theory of concurrency.
Amanda Stent, Assistant Professor, Ph.D., University of Rochester: Spoken and multimodal dialogue systems, natural language generation, theories of discourse, information extraction.
Scott Stoller, Associate Professor, Ph.D., Cornell University: Distributed systems; fault-tolerance and security; software testing and verification; program analysis and optimization.
Michael Tashbrook, Lecturer, M.C.S., University of Virginia: Computer science education; software engineering.
M. Alex O. Vasilescu, Assistant Professor, Ph.D., University of Toronto: Computer vision; computer graphics (animation, image-based rendering); tensor (multilinear) algebra; physics-based modeling; machine learning.
Computer science is the study of computer systems, including the architecture of computers, development of computer software, information processing, computer applications, algorithmic problem-solving, and the mathematical foundations of the discipline.

The Computer Science major provides professional education in computer science to prepare the student for graduate study or for a career in the computing field. Students learn concepts and skills needed for designing, programming, and applying computer systems while also learning the theoretical and mathematical foundations of computer science. They have sufficient freedom in the program to pursue other academic interests in the liberal arts, sciences, and engineering to complement their study of computer science.

Many students prepare for their professional careers through internships at local companies. Computer science graduates are recruited heavily, and career opportunities include developing software systems for a diverse range of applications such as: user interfaces; networks; databases; forecasting; web technologies; and medical, communications, satellite, and embedded systems. Many are employed in the telecommunication and financial industries, and some are self-employed as heads of software consulting companies.

The Department of Computer Science offers two undergraduate majors: Computer Science and Information Systems. Requirements and courses for the latter appear under the program title in the alphabetical listings of Approved Majors, Minors, and Programs. The two programs of study share a number of courses, particularly in the first two years, so that it is possible to follow a program that permits a student to select either major by the start of the junior year. The Department also offers a minor in computer science, a joint B.S./M.S. program, and an honors program.

Program Educational Objectives
Within five years of graduation, alumni of the Computer Science undergraduate program should be:
1. Conducting successful careers in computer science-related disciplines and adapting to emerging markets and technologies.
2. Contributing to the development of local, national, and global economies.
3. Pursuing life-long learning opportunities, particularly graduate education.
4. Leading interdisciplinary design teams in government, academic, or industrial settings.

Program Outcomes
On completion of the program, graduates of the program should be able to:
1. design, develop, test, and evaluate software systems;
2. recognize the need for, and expect to engage in, life-long learning for continued professional excellence;
3. apply their knowledge to the solution of practical and useful problems;
4. communicate effectively; and
5. work collaboratively.

In addition, undergraduates must:
6. have a solid understanding of computational theory and foundational mathematics;
7. have substantial exposure to advanced topics in software and computing systems;
8. have a comprehensive general education background;
9. be prepared to successfully enter the job market and/or graduate studies; and
10. understand professional responsibility.

More details about program educational objectives and outcomes can be found at http://cs.sunysb.edu/admissions/Objectives.html.

Computing Facilities
Computing facilities for undergraduates are maintained by both the University Computing Center and the Department of Computer Science. For a description of the computing services provided by the University Computing Center, see page 44 in the Student Services section of this Bulletin.

The Department of Computer Science provides additional laboratories to support undergraduate instruction and research. The laboratory facilities are regularly upgraded to keep pace with advances in technology. Current computing facilities include the Computer Science Undergraduate Computing Laboratory; the Programming Techniques Teaching Laboratory with facilities for classroom instruction; the Computer Associates Transactions Laboratory, used primarily for upper-level courses on databases, transaction processes, and Web applications; the Computer Science Advanced Programming Laboratory, also donated by Computer Associates, Inc., which provides computing support for upper-level courses on such topics as operating systems and user interfaces; and the Computer Science Multimedia Laboratory, used for courses on multimedia design. Most of the laboratories are connected to the Internet via the campus network and are easily accessible by students from campus residences or from off-campus via modem.

The Departmental research laboratories are available to undergraduate students working on supervised projects with computer science faculty.
Transfer Credits

Students who wish to transfer credits for courses equivalent to CSE 114, 214, or CSE 215 in order to use them as prerequisites for other CSE courses or toward the requirements for acceptance into the major must demonstrate proficiency in the course material by passing a proficiency examination, given during the first week of each semester.

Courses Offered in Computer Science

See the Course Descriptions listing in this Bulletin for complete information.

CSE 101  Introduction to Computers and Information Technologies
CSE 102  Introduction to Web Design and Programming
CSE 110  Introduction to Computer Science
CSE 114  Computer Science I
CSE 130  Introduction to Programming in C
CSE 150  Foundations of Computer Science: Honors
CSE 160  Computer Science A: Honors
CSE 214, 219  Computer Science II, III
CSE 215  Foundations of Computer Science
CSE 220  Computer Organization
CSE 230  Intermediate Programming in C and C++
CSE 260  Computer Science B: Honors
CSE 300  Writing in Computer Science
CSE 301-H  History of Computing
CSE 302  Professional Ethics for Computer Science
CSE 303  Introduction to the Theory of Computation
CSE 304  Compiler Design
CSE 305  Principles of Database Systems
CSE 306  Operating Systems
CSE 307  Principles of Programming Languages
CSE 308  Software Engineering
CSE 310  Data Communication and Networks
CSE 311 Systems Administration
CSE 315 Database Transaction Processing Systems
CSE 320 Computer Architecture
CSE 323 Human Computer Interaction
CSE 325 Computers and Sculpture
CSE 326 Digital Image Processing
CSE 327 Computer Vision
CSE 328 Fundamentals of Computer Graphics
CSE 332 Introduction to Scientific Visualization
CSE 333 User Interface Development
CSE 334 Introduction to Multimedia Systems
CSE 336 Internet Programming
CSE 346 Computer Communications
CSE 350 Theory of Computation: Honors
CSE 352 Artificial Intelligence
CSE 355 Computational Geometry
CSE 364 Advanced Multimedia Techniques
CSE 366 Introduction to Virtual Reality
CSE 370 Wireless and Mobile Networking
CSE 371 Logic
CSE 373 Analysis of Algorithms
CSE 375 Concurrency
CSE 376 Advanced Systems Programming in UNIX/C
CSE 377 Introduction to Medical Imaging
CSE 378 Introduction to Robotics
CSE 380 Computer Game Programming
CSE 381 Advanced Game Programming
CSE 390-394 Special Topics in Computer Science
CSE 408 Network Security

Sample Course Sequence for the Major in Computer Science

**Freshman Fall**  Credits
First Year Seminar 101  1
D.E.C. A  3
CSE 110  3
AMS 151  3
Natural science course  4
Total  14

**Sophomore Fall**  Credits
CSE 214  3
CSE 215  3
AMS 210  3
Natural science course  4
D.E.C.  3
Total  16

**Junior Fall**  Credits
CSE 300  1
CSE Software course  3
CSE 303  3
AMS 310  3
D.E.C.  3
Elective  3
Total  16

**Senior Fall**  Credits
CSE 308  3
CSE Hardware Course  3
CSE elective.  3
D.E.C.  3
Elective  3
Total  15

**Spring Credits**  15
First Year Seminar 102  1
D.E.C.  3
CSE 114  4
AMS 151  3
Natural science course  4
Total  15

**Sophomore Spring**  Credits
CSE 219  3
CSE 220  3
AMS 301  3
D.E.C.  3
D.E.C.  3
Total  15

**Junior Spring**  Credits
CSE 302  1
CSE Software course  3
CSE 373  3
CSE Elective  3
D.E.C.  3
Elective  3
Total  16

**Senior Spring**  Credits
CSE Software course  3
CSE Elective  3
D.E.C.  3
Elective  3
Elective  3
Total  15
CSE 409 Computer System Security
CSE 475 Undergraduate Teaching Practicum
CSE 487 Research in Computer Science
CSE 488 Internship in Computer Science
CSE 495, 496 Senior Honors Research Project I, II

Enrolling in CSE Courses
To enroll in CSE courses, students must:

- Have completed all prerequisites with a grade of C or higher. (Pass/No Credit grades are not acceptable to meet prerequisites.) For transfer students, official transfer credit evaluations must have been completed and approved and the relevant proficiency examination for lower division courses, given during the first week of each semester, must have been taken and passed.
- Failure to satisfy the prerequisites or to attend the first class may result in deregistration. The Pass/No Credit option is not available to CSE majors for CSE courses.

Acceptance into the Computer Science Major
Qualified freshman and transfer applicants may be accepted directly into the Computer Science major upon admission to the University. [Effective fall 2005]
Currently enrolled students may apply for acceptance to the major after completing the following two courses with grades of C or higher.

1. CSE 114 Computer Science I
2. CSE 215 Foundations of Computer Science

Requirements for the Major in Computer Science (CSE)
The major in Computer Science leads to the Bachelor of Science degree. At least five upper-division courses from items 2 and 3 below must be completed at Stony Brook.

Completion of the major requires approximately 80 credits.

1. Required Introductory Courses
   CSE 114 Computer Science I
   CSE 214 Computer Science II
   CSE 215 Foundations of Computer Science
   CSE 219 Computer Science III
   CSE 220 Computer Organization and Systems

2. Required Advanced Courses
   CSE 303 Introduction to the Theory of Computation and CSE 373 Analysis of Algorithms
   CSE 308 Software Engineering
   Three software-related courses chosen from: CSE 305, 306, 304 or 307; 328 or 333
   One hardware-related course chosen from: CSE 310, 320, 346, ESE 345

3. Computer Science Electives
   Three upper-division CSE electives excluding CSE 475, 488, 495, and 496.

4. AMS 151, 161 Applied Calculus I, II
   Note: The following alternate calculus course sequences may be substituted for AMS 151, 161 in major requirements or prerequisites: MAT 125, 126, 127, or MAT 131, 132, or MAT 141, 142 or MAT 171. Equivalency for MAT courses achieved through the Mathematics Placement Examination is accepted to meet MAT course requirements.

5. One of the following:
   MAT 211 Introduction to Linear Algebra
   AMS 210 Applied Linear Algebra
   AMS 326 Numerical Analysis

6. Both of the following:
   AMS 301 Finite Mathematical Structures
   AMS 310 Survey of Probability and Statistics or AMS 311 Probability Theory or AMS 312 Mathematical Statistics

7. One of the following natural science sequences:
   [Effective fall 2005]
   BIO 201, 202, 204 or BIO 201, 203, 204 or BIO 202, 203, 204 Fundamentals of Biology
   or CHE 131, 132, 133 or CHE 141, 142, 143 General Chemistry
   or PHY 131/133, 132/134 or PHY 141, 142 or PHY 125, 126, 127 Classical Physics

8. Four additional credits from the above natural science courses
   [Effective fall 2005]

These courses can be in biology, chemistry, or physics. Advanced natural science courses may be substituted with the prior approval of the Department of Computer Science.

9. Professional Ethics
   CSE 302 Professional Ethics for Computer Science

10. Upper-Division Writing Requirement:
    CSE 300 Writing in Computer Science
    All degree candidates must demonstrate technical writing skills at a level that would be acceptable in an industrial setting. To satisfy the requirement, students must pass CSE 300, a course that requires the completion of various writing assignments, including at least one significant technical paper.
    Note: All students are encouraged to discuss their program with an undergraduate advisor. In Requirement 2 above, CSE/ISE double majors may substitute ESE 440, 441 Electrical Engineering Design I, II for CSE 308 Software Engineering provided that the design project contains a significant software component. Approval of the Department of Computer Science is required.

Grading
All courses taken to satisfy Requirements 1 through 9 must be passed with a letter grade of C or higher. A grade of C or higher is also required in prerequisite courses listed for all CSE and ISE courses.

Specialization in Human-Computer Interaction
The specialization in human-computer interaction emphasizes both the psychology aspects of effective human-computer interactions and the technical design and implementation of systems for those interactions. It requires four core course, two electives, and a project. Students may declare their participation in the specialization after completing the courses in 1a and 1b. All courses must be completed with a grade of C or higher.

1. Core Courses
   a. CSE 323 Human-Computer Interaction
   b. PSY 260 Survey of Cognition and Perception
c. CSE 328 Fundamentals of Computer Graphics or CSE 332 Introduction to Scientific Visualization

d. CSE 333 User Interface Development or PSY 384 Research Lab: Human Factors

2. Two electives from the following, including at least one CSE course:

- CSE 327 Fundamentals of Computer Vision
- CSE 328 Fundamentals of Computer Graphics
- CSE 332 Introduction to Scientific Visualization
- CSE 333 User Interface Development
- CSE 334 Introduction to Multimedia Systems
- CSE 336 Internet Programming
- CSE 352 Artificial Intelligence
- CSE 364 Advanced Multimedia Techniques
- CSE 366 Introduction to Virtual Reality
- CSE 378 Introduction to Robotics
- CSE 390-394 Special Topics in Computer Science*
- PSY 366 Human Problem Solving
- PSY 368 Sensation and Perception
- PSY 369 Special Topics in Cognition and Perception
- PSY 384 Research Lab: Human Factors

*Special topic must be in human-computer interaction.

3. Project

- Completion of CSE 487 Research in Computer Science or CSE 488 Internship in Computer Science or CSE 495/496 Senior Honors Research Project I, II, on a topic in human-computer interaction.

Specialization in Information Assurance

The specialization in information assurance (IA) has been developed as part of the University's establishment of a Center for Cybersecurity and designation by the National Security Agency as a Center of Academic Excellence in Information Assurance Education. This is included in a multifaceted effort to expand and increase information assurance education and research. The specialization deals with the principles, design, development, and management of networks and software systems that provide high levels of assurance in the confidentiality, availability, and integrity of electronic information. It requires four core courses, two electives, and a project. Students may declare their participation in the specialization after completing the courses in 1a and 1b. All courses must be completed with a grade of C or higher.

1. Core Courses

a. CSE 310 Data Communication and Networks or CSE 346 Computer Communications
b. CSE 326 Operating Systems or CSE 376 Advanced Systems Programming in UNIX/C
c. CSE 408 Network Security
d. CSE 409 Computer System Security

2. Two electives from the following:

- CSE 305 Principles of Database Systems
- CSE 306 Operating Systems
- CSE 315 Database Transaction Processing Systems
- CSE 336 Internet Programming
- CSE 375 Concurrency
- CSE 376 Advanced Systems Programming in UNIX/C
- AMS 310 Survey of Probability and Statistics
- AMS 311 Probability Theory
- AMS 312 Mathematical Statistics
- AMS 315 Data Analysis
- AMS 335 Game Theory
- AMS 341 Operations Research I: Deterministic Models
- AMS 342 Operations Research II: Stochastic Models
- EST 412 Intelligence Organizations, Technology, and Democracy

3. Project

- Completion of either CSE 487 Research in Computer Science or CSE 495, 496 Senior Honors Research Projects I, II, on a topic in information assurance.

The Honors Program in Computer Science

The Honors Program in Computer Science, a highly selective academic program within the major in Computer Science, offers a specially designed curriculum to a limited number of exceptional students. The program is open to freshmen and to continuing students. To be admitted as a freshman, students must demonstrate overall academic excellence by achieving a combined SAT score of 1300, an unweighted high school average of at least 93, and high grade averages in mathematics and the natural sciences. Continuing Computer Science majors who have completed at least three CSE courses and have maintained a cumulative grade point average of 3.50 and an average of 3.50 in CSE courses may apply for admission to the honors program in the sophomore or
junior year. Continued participation in the program requires that students maintain a grade point average of 3.50, both cumulative and in all CSE courses.

Honors course offerings include introductory course sequences in programming and in the foundations of computing, advanced courses on selected topics that reflect active research areas within the Department, and a two-semester senior honors project. Students will be able to take at least one honors course each semester throughout a four-year program of study. Honors program students must complete the regular requirements of the Computer Science major, but up to two-thirds of the required computer science courses (see items 1 to 3 in the above list of Requirements for the Major) can be covered by honors courses. Final conferral of honors is contingent upon successful completion of all required courses in the Computer Science major including a minimum of three honors courses, plus the two-semester honors project, with a cumulative grade point average of 3.50 and an average of 3.50 in all CSE courses.

Honors students with a grade point average of 3.50 at the end of the junior year will be automatically approved for admission to the five-year joint B.S./M.S. program in Computer Science. Students who successfully complete the honors program and who decide to enroll in the B.S./M.S. program will be considered for a tuition waiver in the fifth year as well as for a graduate student assistantship. (It is recommended that these students complete an undergraduate teaching practicum in the junior or senior year.)

5. Three upper-division CSE courses totaling at least nine credits (excluding CSE 300, 475, 487, 488).

Note: All of these courses must be passed with a letter grade of C or higher.

Joint B.S./M.S. Program in Computer Science

Computer Science majors may apply for admission to a special program that leads to a Bachelor of Science degree at the end of the fourth year and a Master of Science degree at the end of the fifth year. Students usually apply to the program in their junior year.

Students must satisfy the respective requirements of both the B.S. degree and the M.S. degree, but the main advantage of the program is that six credits may be simultaneously applied to both the undergraduate and graduate requirements. The M.S. degree can therefore be earned in less time than that required by the traditional course of study.

For more details about the B.S./M.S. program, see the undergraduate or graduate program director in the Department of Computer Science.

Requirements for the Minor in Computer Science (CSE)

The minor in Computer Science is open to all students not majoring in either Computer Science or Information Systems or minoring in Information Systems. To declare the minor in Computer Science, students must complete CSE 113 and 114 with grades of C or higher. The minor requires seven CSE courses totaling 22 to 24 credits as outlined below.

1. CSE 114 Computer Science I
2. CSE 214 Computer Science II
3. CSE 215 Foundations of Computer Science
4. CSE 219 Computer Science III or CSE 220 Computer Organization and Systems Programming
5. Three upper-division CSE courses totaling at least nine credits (excluding CSE 300, 475, 487, 488).

Note: All of these courses must be passed with a letter grade of C or higher.